WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Tuesday, November 16, 2004

Hide?	Set Name	Query	Hit Count
		DB=USPT,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR	
☐	L7	L6 same restor$7	3
☐	L6	(sav$4 near3 (hardware or device or machine) near3 configur$9 near3 (setting or state))	12
☐	L5	(sav$4 near2 (hardware or device or machine) near2 configur$9 near2 (setting or state))	6
☐	L4	L2 with restor$5	8
☐	L3	L2 same restor$5	15
☐	L2	(sav$4 near2 configur$9 near2 state)	47
☐	L1	(configur$9 near2 state)	9028

END OF SEARCH HISTORY

L7: Entry 2 of 3 File: USPT Jun 5, 2001

DOCUMENT-IDENTIFIER: US 6243831 B1
TITLE: Computer system with power loss protection mechanism

Detailed Description Text (17):
In the S3 sleeping state, all device configurations are saved in RAM by the operating system, and the operating system directs device drivers and BIOS to turn off all devices. Then, RAM is reduced to a lower power state. Upon return from S3, full power is restored to RAM, all devices are turned back on and reconfigured to the state they were in before S3 was entered by the device drivers and BIOS.

Detailed Description Text (18):
The S4 sleeping state is the lowest power state because RAM is turned off, in contrast to the S1, S2, and S3 sleeping states, where RAM continues to be powered. In the S4 sleeping state, all device configurations are saved to RAM under the direction of the operating system. Then, the operating system saves RAM to a hibernation file somewhere in a non-volatile storage medium. The operating system then powers down the computer system S, except for trickle current, in some cases. Resume from the S4 state results from a variety of events, dependent upon designer choices. For example, an incoming fax or phone call may trigger a return from S4. When resume occurs, the contents of the hibernation file are restored to RAM by the operating system, and all devices are re-powered and reconfigured to the pre-S4 state.

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#
End of Result Set

Generate Collection Print

L7: Entry 3 of 3 File: USPT Dec 31, 1996

DOCUMENT-IDENTIFIER: US 5590315 A
TITLE: Method and apparatus for simulating user input device presence in a computer system

CLAIMS:

8. A method of preventing the loss of device configuration on standby in a computer system including a processor, a first device port capable of being powered down for coupling to a first physical user input device and controller circuitry for communicating between the processor and the first device port and for receiving device configuration command characters from the processor, the configuration command characters being suitable for altering the configuration of the first physical user input device, which in turn is configurable by the configuration command characters but is incapable of providing its configuration state to the controller circuitry, the method comprising the steps of:

(a) capturing in the computer system predetermined device configuration command characters directed to the first device port from the processor;

(b) saving in the computer system a device configuration state of the physical user input device as indicated by the device configuration command characters captured in step (a);

(c) powering down the device port;

(d) powering up the device port; and

(e) resending said device configuration command characters responsive to the device configuration state saved in the computer system at step (b) to the first device port to restore the device configuration state to the physical user input device.

Previous Doc Next Doc Go to Doc#

L5: Entry 1 of 6 File: USPT Aug 10, 2004

DOCUMENT-IDENTIFIER: US 6775728 B2
TITLE: Method and system for concurrent handler execution in an SMI and PMI-based dispatch-execution framework

Detailed Description Text (41):
As discussed above, SMM Nub 24 is responsible for coordinating activities while the processors are operating in SMM. The various functions and services provided by one embodiment of SMM Nub 24 are graphically depicted in FIG. 7. These functions and services include synchronizing all of the processors for multiprocessor configurations, saving the machine state, including floating point registers, if required, and flushing the cache, as provided by function blocks 134, 136, and 138. The SMM Nub also provides a mode switching function 140 that switches the processor mode from real mode to protected mode, as discussed above with reference to block 130. Mode switching function 140 also enables the processor's internal cache. Other functions provided by SMM Nub 24 include setting up a call-stack in SMRAM 26, maintaining list of handlers 46, and maintaining handler queue 48, as depicted by function blocks 142, 144, and 146.

L4: Entry 1 of 8 File: USPT Jul 13, 2004

DOCUMENT-IDENTIFIER: US 6763327 B1
TITLE: Abstraction of configurable processor functionality for operating systems portability

CLAIMS:

2. A software library as in claim 1, wherein the areas of processor configurability include save and restore of configurable processor state.

14. A method as in claim 13, wherein the first and second configurations include parameters for processor configurability, the parameters including two or more of save and restore of configurable processor state, co-processor initialization and control, interrupts, timers, instruction caches, processor data caches, processor debug features, and instruction stream disassembly.

15. A method of designing a configurable processor, the configurable processor having a first instruction set architecture portion that is not configurable and a second instruction set architecture portion that is user-configurable, the configurable processor being able to execute software intended for a desired operating system, the method comprising: automatically providing a common software interface to a plurality of different operating systems, including the desired operating system; receiving a desired configuration for the configurable processor, the desired configuration specifying parameters for configuration of the second instruction set architecture portion of the configurable processor, the parameters including two or more of save and restore of configurable processor state, co-processor initialization and control, interrupts, timers, instruction caches, processor data caches, processor debug features, and instruction stream disassembly; automatically generating an abstraction layer based on the received configuration; and building a software implementation using the common software interface and the abstraction layer, the software implementation being executable together with the desired operating system on the configurable processor.

16. A system for designing a configurable processor, the configurable processor having a first instruction set architecture portion that is not configurable and a second instruction set architecture portion that is user-configurable, the configurable processor being able to execute software intended for a desired operating system, the system comprising: a common software interface to a plurality of different operating systems, including the desired operating system; a user interface for receiving a desired configuration for the configurable processor, the desired configuration specifying parameters for configuration of the second instruction set architecture portion of the configurable processor, the parameters including two or more of save and restore of configurable processor state, co-processor initialization and control, interrupts, timers, instruction caches, processor data caches, processor debug features, and instruction stream disassembly; means for automatically generating an abstraction layer based on the received configuration; and means for building a implementation using the common software interface and the abstraction layer, the software implementation being executable together with the desired operating system on the configurable processor.

L4: Entry 3 of 8 File: USPT May 8, 2001

DOCUMENT-IDENTIFIER: US 6230259 B1
TITLE: Transparent extended state save

CLAIMS:

1. A microprocessor, comprising:

a microprocessor core, including:

a standard register file; and

an extended register file;

wherein said microprocessor core is configured to execute standard instructions that use said standard register file, but not said extended register file;

wherein said microprocessor core is further configured to execute extended instructions that use said extended register file; and

wherein said microprocessor core is further configured to receive interrupts or exceptions; and

an extended state save circuit coupled to said microprocessor core, wherein said extended state save circuit is configured to restore, in response to a return from an interrupt or exception, a state of the extended register file for a process using said extended instructions if said interrupt or exception was received during execution of said process.

2. The microprocessor as recited in claim 1, wherein said extended state save circuit is further configured to restore said state from a saved version of said state if another process using said extended instructions was executed after said interrupt or exception and before said return.

3. The microprocessor as recited in claim 2, wherein said extended state save circuit is further configured to restore said state as the current state of said extended register file if no other process using said extended instructions was executed after said interrupt or exception and before said return.

7. The microprocessor as recited in claim 6, wherein said extended state save circuit is configured to compare the contents of said buffer identification register to the saved identifier for the process being executed when the interrupt or exception was received and restore the state of the extended register file from the memory buffer identified by the saved identifier if the saved identifier differs from the contents of said buffer identification register upon return from said interrupt or exception.

16. The microprocessor as recited in claim 1, wherein said microprocessor core is configured to execute an instruction for returning to a process using said extended instructions, wherein in response to said instruction for returning to a process using said extended instructions, said extended state save circuit is configured to

L4: Entry 4 of 8 File: USPT Oct 24, 2000

DOCUMENT-IDENTIFIER: US 6138194 A
TITLE: Apparatus for sensing movement of a bus card and automatically removing power from the bus card

Brief Summary Text (14):
In another embodiment of the present invention, the controller is configured to detect an insertion of the bus card into the bus connector by monitoring the movement sensor, and to reset the bus card upon detection of the insertion. In a variation on this embodiment, the controller is configured to reset the bus card to an initial state. In another variation, the controller is configured to save state from the bus card to the computer system before removing power from the bus card, and to reset the bus card by restoring the state from the computer system to the bus card.

CLAIMS:

1. An apparatus for removing connections to a bus card in a computer system when the bus card is inadvertently removed from the computer system while the computer system is operating, comprising:

a bus connector, for receiving the bus card;

a power conductor coupled to the bus connector, to provide an electrical coupling between the bus card and a power source;

a power switch coupled between the power conductor and the bus connector, to selectively provide power to the bus card in the bus connector;

a movement sensor positioned to be in communication with the bus card when the bus card is mounted in the bus connector, for sensing a movement of the bus card from the bus connector; and

a controller coupled between the movement sensor and the power switch, for activating the power switch to remove power from the bus card when the movement sensor detects the movement of the bus card from the bus connector;

wherein the controller is configured to detect an insertion of a new bus card into the bus connector by monitoring the movement sensor, and to reset the new bus card upon detection of the insertion; and

wherein the controller is configured to save a state from the bus card to the computer system before removing power from the bus card, and to reset the new bus card by restoring the state from the computer system to the new bus card.

11. An apparatus for removing connections to a bus card in a computer system when the bus card is inadvertently removed from the computer system while the computer system is operating, comprising:

a bus connector, for receiving the bus card;